Turbomole 5.8 ja uudet ominaisuudet

Mikael Johansson
mikael.johansson@csc.fi

Turbomolen uusin versio asennettiin äskettäin IBMSC:lle sekä uudelle alustalle, Sepelille. Turbomole on kvanttikemian ohjelmisto, jonka perinteiset vahvuudet ovat olleet Hartree–Fock ja tiheysfunktionaalimenetelmien (DFT) tehokas implementaatio. Myös alemman tason korreloidut aaltofunktiomenetelmät kuten MP2 ja CC2 ovat jo jonkin aikaa kuuluneet pakettiin.

Artikkelissa käymme lyhyesti läpi uuden version sisältämiä pääuutuuksia:

- **MARIJ-DFT** rinnakkaistettu. Nk. puhtailla DFT-funktionaaleilla laskeminen on Turbomolella erityisen nopeaa. Tätä vieläkin tehokkaampi on multipolikiihdytetty versio RI-DFT:stä, MARIJ-DFT. Erityisesti kookkaille systeemeille MARIJ-DFT voi olla huomattavasti nopeampi kuin RI-DFT.
- **NumForce** toimii kunnolla rinnakkaisesti. Numeerisia IR-spektrejä nopeasti.
- **RI-CC2** ja **RI-MP2** rinnakkaistettu. Nyt näitäkin voidaan ajaa tehokkaasti usealla prosessorilla.
- **ECP**-pseudopotentiaalit g-projektioin. Nyt esim. lantanidien käsittely on mahdollista Turbomolessa, ainakin jossain määrin

Turbomolen peruskäyttöä ei tässä käsitellä. Vuoden 2006 aikana on kuitenkin suunnitteilla Turbomole-kurssi sekä aloittelijoille että pidemmälle ehtineille. CSC:n Turbomole-sivustolta löytyy myös aikaisempaa, noviiseille suunnattua kurssimateriaalia.

IBMSC:n ja Sepelin erot ja yhtenäisyydet

Turbomole 5.8 löytyy siis sekä IBMSC:ltä että Sepeliltä. Ohjelman käyttö poikkeaa hieman riippuen koneesta. Molemmissa kuitenkin Turbomolen interaktiivinen käyttö alustetaan komennolla

```
use turbo58
```

Tämän jälkeen esimerkiksi alustusohjelma **define** on käytössä. Turbomole-spesifiset tiedostot, **control** ja kumppanit, toimivat melkein ilman muutoksia molemmissa koneissa. Mikäli siirtää tiedostoja koneiden välillä, pitää kuitenkin muistaa $parallel_platform rivin poistaminen controlista, muutoin rinnakkaisajo on tehottomampaa.

Eräajotiedostot ja niiden jonoonlähettämiskäytännöt ovat kuitenkin konespesifisiä, jonojärjestelmien ollessa erilaiset. CSC:n Turbomole-sivustolta löytyy esimerkkejä eräajotiedostoista.

Koneiden välillä on myös toinen ero: rinnakkaisissa Turbomole-ajoissa IBMSC:llä on pakko aina varata yksi prosessi enemmän kuin mitä ohjelma voi käyttää, eli työskriptissä pitää asettaa PARNODES esimerkiksi seuraavasti:

```
export PARNODES=$(( `echo $LOADL_PROCESSOR_LIST | wc -w` -1 ))
```

Sepelissä sen sijaan tätä rajoitusta ei ole ja voidaan käyttää kaikkia prosessoreita laskemiseen, eli esimerkiksi:

```
export PARNODES=$NSLOTS
```

Eroja löytyy tietenkin myös tehokkuudessa. Sepelin prosessorit ovat nopeampia kuin IBMSC:n. Sepelissä on myös paikallista levyä, joka jaetaan korkeintaan neljän työprosessin kesken; tätä hyödyntäen levytoiminnot nopeutuvat huomattavasti. Yhteydenpito prosessien välillä on puolestaan tehokkaampaa IBMSC:ssä, joten jotkut moduulit rinnakkaistuvat hieman paremmin kuin Sepelissä.

MARIJ-DFT, nopea, mutta...

Multipole Accelerated Resolution of the Identity J (Coulomb)-DFT, eli **MARIJ-DFT** on Turbomolen versio melkein lineaarisesti systeemin koon mukaan skaalautuvasta DFT-implementaatiosta. Skaalautuvuus on parhaimmillaan $N^{1,5}$ luokkaa. MARIJ-DFT toimii RI-DFT:n tavoin vain puhtailla funktionaaleilla jotka eivät sisällä Hartee–Fock vaihtoa. Tällöin voidaan tehokkaasti hyödyntää nk. tiheyssovitusta, RI-approksimaatiota. Tämä toimii Turbomolessa **LDA**:n lisäksi GGA-funktionaaleilla **BLYP, BP86** ja **PBE**, sekä uudella meta-GGA:lla **TPSS**.

MARIJ-DFT:n käyttöönotto on helppoa. Tavallisen RI-DFT laskun määrittelemisen lisäksi on **define**:n viimeisestä ruudusta, eli "GENERAL MENU":sta valittava optio **marij**. Tällöin define näyttää valikon, josta halutessa voi säätää MARIJ-parametreja. Oletusarvot toimivat yleensä hyvin. **control**-tiedostoon ilmestyy avainsana $marij.

Taulukossa 1 on lueteltu ajoaikoja esimerkkilaskulle. Taulukosta nähdään ensinnäkin, että RI-approksimaatio nopeuttaa laskuja huomattavasti. Sama lasku kestää tavallisella DFT:llä kahdeksalla prosessorilla yli kaksi kertaa kauemmin kuin RI-DFT yhdellä CPU:lla. MARIJ-DFT nopeuttaa laskua vieläkin enemmän ollen yhdellä CPU:lla noin 40% nopeampi kuin RI-DFT.

MARIJ-DFT skaalautuu kuitenkin prosessorien määrään katsoen melko huonosti. Rinnakkaisajoissa RI-DFT:n ajoaika lähenee nopeasti MARIJ-DFT:n aikaa. Sepelillä molemmat ovat melkein yhtä nopeita jo 8 CPU:lla, IBMSC:llä noin 16 CPU:lla. MARIJ-DFT on kuitenkin aina nopeampi kuin pelkkä RI-DFT, eli mikäli erityistä syytä ei ole, kannattaa aina käyttää MARIJ-DFT:tä.

(Taulukosta puuttuu kahden CPU:n rinnakkaisajot IBMSC:lle; muistetaan että yksi CPU menee "hukkaan" IBMSC:n TURBOMOLElla.)

Ajotyyppi	CPU:t	Ajoaika, Sepeli	Ajoaika, IBMSC
ei RI:tä (dscf)	8	14:13 h	27:42 h
tavallinen RI-DFT	1	7:02 h	11:24 h
	2	5:58 h	-
	4	3:40 h	7:35 h
	8	2:47 h	4:03 h
	16	-	3:12 h
MARIJ-DFT	1	4:31 h	6:48 h
	2	3:42 h	-
	4	2:57 h	5:46 h
	8	2:35 h	3:34 h
	16	-	3:08 h

Taulukko 1. Ajoaikoja BP86-funktionaalilla: 30 SCF-iteraatiota, 323 atomia ja 1.271 elektronia, SVP kantajoukko (yht. 3.156 funktiota), UKS, "ricore" 500 MB.

Sekä RI-DFT että MARIJ-DFT nopeutuvat yleensä huomattavasti jos algoritmeilla on paljon muistia käytössään. Muistin määrää säädellään avainsanalla $ricore. Optimaalisen määrän löytäminen voi olla hieman hankalaa, mutta 700 MB on hyvä kompromissi . Tämä asetetaan **control**-tiedostoon seuraavasti:

```
$ricore 700
```

Muistimäärän lisääminen on myös huomioitava jonojärjestelmän resurssipyynnöissä! Esim. tuon yllä olevan 700 MB:n lisäksi prosessit käyttävät muistia myös muuhun; mitä isompi lasku, sen enemmän oheismuistia kuluu. 2 GB:n varaus prosessia kohden ei ole yhtään liioiteltua.

Sekä RI-DFT että MARIJ-DFT -laskuja voidaan edelleen nopeuttaa käyttämällä suppeampaa RI-kantaa SCF-iteraatioiden ajan. Tämä asetetaan **define**:stä käsin valitsemalla trunc, joka lisää avainsanan $truncated RI. Tätä käytettäessä on (kuten aina) tarkistettava että SCF-lasku on konvergoinut oikeaan elektroniseen tilaan esim. apuohjelmalla **eiger**.

Värähtelyspektrejä NumForce:lla

Vaikka MARIJ-DFT ei skaalaudu kovin hyvin CPU-määrän suhteen, on sen nopeudesta kuitenkin huomattavaa iloa töissä, joissa on laskettava monta erillistä yksiprosessorilaskua. Värähtelyspektrin (IR) laskeminen numeerisesti on tyyppiesimerkki tästä. Usein kannattaa laskea spektri analyyttisesti, TURBOMOLEssa ohjelmalla **aoforce**. Tämä ei kuitenkaan aina ole mahdollista. Tarvittavat analyyttiset toiset derivaatat löytyvät ainoastaan **s, p ja d** kantafunktioille; aoforce ei ymmärrä f-funktioita. Systeemi voi myöskin olla niin suuri, ettei aoforce pysty sitä käsittelemään lainkaan, tai lasku olisi ei-rinnakkaisuuden vuoksi äärimmäisen hidas. Myöskään virittyneitä tiloja ei tällä hetkellä voida käsitellä analyyttisesti. Tällöin vibraatiospektri on laskettava numeerisesti.

TURBOMOLEssa numeerinen spektri lasketaan **NumForce**-ohjelmalla. NumForce voidaan ajaa rinnakkaisesti, mikä tarkoittaa, että se käynnistää annetun CPU-määrän verran yksi-prosessorilaskuja optimoidusta rakenteesta hieman poikkeaville geometrioille. Tämä ei ole varsinaisesti version 5.8 uusi ominaisuus; uutta on että se toimii. Seuraava komento käynnistäisi laskun 42 prosessorilla, esimerkissä RI-approksimaatiolla (vaikkapa MARIJ-DFT tai RI-MP2):

```
NumForce -ri -np 42 > numforce.out
```

Skaalautuvuus CPU-määrän suhteen on luonnollisesti erittäin hyvä.

Vinkkinä kerrottakoon, että numeerinen spektri vaatii hyvin konvergoituneen energian SCF-laskuille. NumForce asettaa automaattisesti $scfconv-parametrin arvoksi vähintään 7, mutta sitäkin tiukempi konvergenssikriteeri vaikuttaa vielä aika paljon tarkkuuteen; kannattaa käyttää:

```
$scfconv 8
```

Rinnakkainen RI-CC2 ja optiset spektrit

Tavallinen menetelmä virittyneiden tilojen ominaisuuksien laskemiseen on aikariippuvainen TDDFT. Usein TDDFT antaa halvalla hyvää; sen myötä optisten spektrien (UV/VIS) laskeminen on tullut mahdolliseksi jopa suurille biomolekyyleille. TDDFT ei kuitenkaan ole ongelmaton menetelmä. Nykyisten funktionaalien puutteista johtuen anioneita tai eksitaatioita, jotka johtavat elektronitiheyden huomattavaan siirtymiseen molekyylissä ei voida kuvata TDDFT:llä.

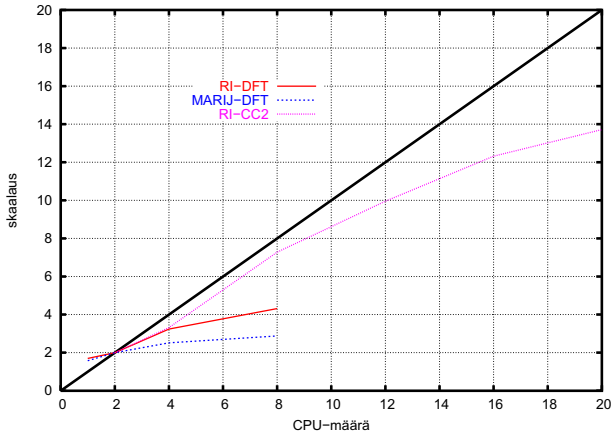
Kun TDDFT ei toimi voidaan käyttää erilaisia korreloituja aaltofunktiomenetelmiä. Nämä ovat yleensä hyvin raskaita. RI-approksimaatioita voidaan kuitenkin hödyntää tehokkaasti myös tässä. Ab initio-menetelmien kirjosta CC2:sta (approksimaatio CCSD Coupled Cluster Singles and Doubles-menetelmästä) on RI:n kautta tullut varteenotettava vaihtoehto TDDFT:lle. Rinnakkaistettuna se on nyt entistäkin varteenotettavampi. Tällä hetkellä rinnakkaisesti voi laskea perustilan energian ja mielenkiintoisempana elektronisen viritysspektrin. CC2:lla on esimerkiksi tutkittu lehtivihreän valoabsorptiota, mikä on tyypillinen TDDFT:n kompastuskivi.

RI-CC2 skaalautuu erittäin hyvin prosessorien määrän suhteen. Esimerkkinä erään piivedyn spektrin referenssilasku hyvin suurella kantajoukolla, ajoaikoja taulukossa 2. Skaalautuminen on hetkittäin jopa superlineaarista; kaksinkertaisella prosessorimäärällä ajo joskus kestää alle puolet ajasta.

CPU-määrä	ajoaika
2	220:51 h
4	133:14 h
8	60:54 h
12	44:13 h
16	35:49 h
32	24:49 h

Taulukko 2. Sepelin ajoaikoja Si_6H_{14} -molekyylin spektrilaskusta. RHF, aug-cc-pV(Q+d)Z kantajoukko piille, aug-cc-pVQZ vedylle, yhteensä 1.178 funktiota. Symmetria C_{2h}; jokaiselle neljälle symmetriaryhmälle laskettiin alimmat 8 siirtymää, yhteensä 32. Kaikki elektronit korreloituja.

HUOM! IBMSC:llä ei kannata laskea RI-CC2:sta. Laskut ovat hyvin levyintensiivisiä, kirjoittaen ja lukien useita gigatavuja tiedostoja prosessia kohden. IBMSC:llä ylläoleva ajo ei ehtinyt valmistua ennen kuin jonon määräaika umpeutui. Tarkastelu yhdestä laskun ensimmäisistä askelista (osa CCS-alustusoptimoinnista) on kuvaavaa. Kahdeksalla prosessorilla IBMSC on jo 3,5 kertaa hitaampi kuin Sepeli. Tilanne pahenee entisestään prosessorimäärän nostolla 16:sta. Kaksinkertaisella CPU-määrällä ajoaika kasvaa yli kaksinkertaiseksi IBMSC:n levyjärjestelmän tukkeutuessa täysin.



Kuva 1. RI-DFT, MARIJ-DFT ja RI-CC2 -skaalautuvuus sepelillä. Perusyksikkönä on käytetty kahden prosessorin rinnakkaisajoa.

Levyintensiivisyyden takia myös Sepelillä on pakko käyttää paikallista levytilaa väliaikaistiedostoille. Tämä on erikseen määriteltävä, joko **define**:n kautta tai sitten suoraan editoimalla **control**-tiedostoa. Kummin päin tahansa, seuraavat avainsanat on löydyttävä controlista (polku sopivasti muutettuna):

```
$TMPDIR /tmp/erkkiesimerkki/ricc2-tiedostoja
$SHAREDTMPDIR
```

Tätä kirjoittaessa manuaalissa on virheellisesti lueteltu yllä olevat avainsanat pienillä kirjaimilla.

RI-CC2 tarvitsee ja osaa hyödyntää runsaasti RAM-muistia, aivan kuten muutkin RI-laskut. Esimerkkilaskussa oli varattu 2GB:tä muistia itse RI-osalle määrittämällä avainsana:

```
$maxcor 2048
```

Enempää ei oikein ole mahdollista varata nykyisillä alustoilla.

Hyvistä puolistaan huolimatta, CC2 ei käsittele elektronikorrelaatiota kovinkaan kattavasti. Se on riittämätön erityisesti multikonfiguraatiomolekyyleille ja virityksille, jotka eivät ole yksöiseksitaation hallinnoimia Tämäntapaisten korkeamman tason ab initio-menetelmiä vaativien ongelmien ratkomiseen CSC:lle äskettäin asennettu Molpro-ohjelmisto puolestaan on omiaan.

MP2-energioita rinnakkaisesti ricc2-ohjelmalla

Myös MP2, toisen asteen Møller–Plesset häiriöteoria, on rinnakkaistettu TM 5.8:ssa. MP2 on mm. halvin kvanttikemiallinen menetelmä, joka osaa huomioida dispersio- eli van der Waalsin -voimia, johon esim. nyky-DFT ei pysty.

RI-MP2 on rinnakkaistettu, hieman harhaanjohtavasti, **ricc2**-moduulissa, ei **rimp2**-moduulissa (ricc2 osaa myös laskea esim. CCS ja CIS-tasolla). Laskentataso voidaan valita definen uudesta cc2-menusta käsin tai editoimalla $ricc2-ryhmää controlissa. MP2-taso valitaan lisäämällä ryhmään mp2, eli lyhimmillään:

```
$ricc2
        mp2
```

Mikäli halutaan tietää ainoastaan MP2-energia eikä esim. erilaisten diagnostiikkaparametrien arvoja, voidaan MP2-laskua nopeuttaa jopa 75% näin:

```
$ricc2
        mp2 energy only
```

MP2-lasku skaalautuu hyvin CPU-määrän suhteen. Sekä MP2-laskuun että sitä pakollisena edeltävään Hartree–Fock **dscf**-laskuun voi hyvin käyttää ainakin 24 prosessoria. Näistä Hartree–Fock -osa on aikaa vievin askel, tyypillisesti kestäen yli viisi kertaa kauemmin kuin MP2-energian laskeminen. MP2 vaatii tosin suuremman kantajoukon kuin normaali HF-lasku. Kaksoispolarisoitu **triple-zeta** -kanta kuten TZVPP tai cc-pVTZ tarjoaa hyvän hinta/laatu-suhteen.

Rinnakkaiset MP2-laskut käyttävät levyä ja tilapäistiedostoja vastaavalla tavalla kuin CC2-laskut, eivät tosin yhtä paljoa. Kts. edellinen kappale.

Pseudopotentiaalit g-projektioilla

Tätä on tietyissä piireissä odotettu kauan; tehokasta Turbomolea ei ole optimaalisesti voitu käyttää yhdisteille jotka sisältävät raskaita alkuaineita, kuten lantanideja. Syynä on ollut ECP-pseudopotentiaalien rajoittuminen f-projektioihin, joka ei aina riitä. Nyt g-projektiot ovat vihdoin käytettävissä. Homma ei kuitenkaan ole ihan yhtä suoraviivaista kuin yksinkertaisemmille pseudopotentiaaleille. Ensimmäinen niksi joka pitää tietää on erään avainsanan lisääminen **control**-tiedostoon:

```
$newecp
```

Ilman tuota Turbomole käyttää vanhaa ECP-koodia, joka tuottaa virheilmoituksen liian edistyneiden pseudojen osalta. Käyttöön liittyy muutakin elämää hankaloittavaa. Suurimpana epämukavuutena voidaan pitää Extended Hückeliin perustuvan orbitaalimiehityksen alkuarvauksen parametrien puuttuminen; eri orbitaalisymmetrioiden miehitykset on keksittävä itse (eli kokeiltava tai muutoin hankittava).

Turbomolen standardikirjastosta ei myöskään vielä löydy kovin montaa g-projektioin varustettua kantajoukkoa. Näitä on kuitenkin helppo lisätä itse, kantajoukkoja löytää esimerkiksi Pacific Northwest Laboratoryn ylläpitämän Gaus-

sian Basis Set Order Form:in kautta (joka jopa osaa tulostaa kantajoukot TURBOMOLE-formaatissa). CSC:lle on valmiiksi asennettu Caon ja Dolgin vuoden 2001 ECP:t lantanideille. Sekä kantajoukko että itse ECP löytyvät kirjastosta nimellä ecp-28-mwb-2001.

Yhteenvetona

- Sepeli on useimmissa tapauksissa soveliaampi alusta TM-laskuille kuin IBMSC.
- Jos voit käyttää RI-DFT:tä, käytä samalla MARIJ-DFT:tä!
- Kannattaa hyödyntää NumForce:n lähes optimaalista rinnakkaistumista.
- CC2 voi olla etsimäsi vaihtoehto TDDFT:lle optisten spektrien tutkimisessa.
- MP2-energiat saa nyt melkein HF-laskun kaupanpäällisinä.
- Pseudopotentiaalien g-projektiot toimivat.

WWW-linkkejä

- CSC:n TURBOMOLE-sivusto: http://www.csc.fi/chem/progs/turbomole.phtml.en
- TURBOMOLEn uudet kotisivut: http://www.cosmologic.de/turbomole.html
- Gaussian Basis Set Order Form: http://www.emsl.pnl.gov/forms/basisform.html
- CSC:n MOLPRO-sivusto: http://www.csc.fi/chem/progs/molpro.phtml.en

Lisätietoja

TURBOMOLEa koskevia kysymyksiä sekä palautetta voi osoittaa allekirjoittaneelle osoitteella mikael.johansson@csc.fi, tai puhelimitse numeroon 09-457 2934.

Ei-koodaavien RNA-sekvenssien tunnistaminen Rfam-tietokannan avulla

Kimmo Mattila
Kimmo.Mattila@csc.fi

Ei-koodaavan RNA:n tutkimus on sekä kokeellisessa biologiassa että bioinformatiikassa edennyt viime vuosina nopeasti. Ei-koodaavan RNA:n geenejä eli sellaisia, joiden lopputuote ei ole proteiini vaan toiminnallinen RNA-molekyyli, tunnetaan nykyään kaikista eliöryhmistä. Esimerkki ei-koodaavasta RNA:sta on ribosomaalinen-RNA. Uusia RNA-geenejä tunnistetaan jatkuvasti ja samalla on syntynyt tarve koota ja käsitellä ei-koodaavaa RNA-dataa bioinformatiikan keinoin, esimerkiksi genomeja annotoitaessa.

Proteiinien tapaan ei-koodaavan RNA:n etsinnässä ja analysoinnissa voidaan käyttää hyväksi toiminnaltaan samankaltaisten sekvenssien konservoituneisuutta. Myös ei-koodaavalle RNA:lle tunnetaan samankaltaisten sekvenssien muodostamia perheitä, joita voidaan kuvata proteiineille käytettäviä HMM-profiileja muistuttavilla menetelmillä. Toisin kuin proteiinien kohdalla ei-koodaavien RNA-sekvenssien analysoinnissa sekundäärirakenteet ovat keskeisessä asemassa, joten profiileissa käytetään menetelmiä, joissa huomioidaan sekä sekvenssin että sekundäärirakenteen konservoituneisuus.

Yleisimpiä RNA-sekundäärirakenneprofiilien esittämiseen käytettyjä menetelmä ovat kovarianssimallit (covariance models), jotka perustuvat stokastisiin kontekstittömiin kielioppeihin. Kovarianssimallien käytöllä on kuitenkin rajoitteita. Ne eivät pysty käsittelemään sekundäärirakenteessa mahdollisesti olevia pseudosilmukoita (pseudoknot). Tämän puutteen käytännön merkitystä ei vielä tunneta kovin hyvin, mutta sen oletetaan vaikuttavan vain harvoissa tapauksissa. Toinen ongelma on kuitenkin hyvin konkreettinen: kovarianssimallit ovat laskennallisesti hyvin raskaita.

Infernal-ohjelmisto

Infernal-ohjelmisto [1] tarjoaa työkaluja kovarianssimalleihin perustuvien RNA-sekundäärirakenneprofiilien luomiseen ja niillä tehtävään sekvenssianalyysiin. Ohjelma on kehitetty Sean Eddyn laboratoriossa (Washington University in St. Louis) eli siis samassa ryhmässä, josta HMMER-ohjelmisto on peräisin. Infernal-ohjelmisto sisältää työkaluja, joilla voidaan luoda uusia RNA-profiileja, rinnastaa sekvenssejä RNA-profiiliin tai etsiä sekvenssistä tai sekvenssitietokannasta annettua RNA-profiilia vastaavia kohtia. RNA:n sekundäärirakenteen ennustustyökaluja Infernal-ohjelmisto ei sisällä. CSC:llä on tätä varten käytettävissä Mfold- ja Vienna-ohjelmistot.

Infernal-ohjelmistoa kehitetään jatkuvasti ja kehittäjänsä mukaan se on vielä hyvin keskeneräinen.
Esimerkiksi tietokantahauissa käytettävään cmsearch-ohjelmaan ei ole vielä luotu menetelmiä, joilla voitaisiin laskea BLAST- ja HMMER-ohjelmistojen tapaan osuman tilastollista merkitystä kuvaava E-arvo. Siitä huolimatta ohjelmistoa käytetään jo nyt laajalti.

Rfam-tietokanta

Siinä missä Infernal on analoginen HMMER-ohjelmistolle, Rfam-tietokanta [2] on RNA-maailman vastine Pfam-proteiniperhetietokannalle. Sanger-instituutissa ylläpidetty Rfam-tietokanta sisältää huolellisesti analysoituja ei-koodaavan